(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 706 118 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
01.03.2000 Bulletin 2000/09

(21) Application number: **95916035.9**

(22) Date of filing: **21.04.1995**

(51) Int Cl.[7]: **G06F 9/06**, G06F 1/00

(86) International application number:
PCT/JP95/00796

(87) International publication number:
WO 95/29438 (02.11.1995 Gazette 1995/47)

(54) PROGRAM DATA PROTECTING METHOD

METHODE ZUR PROGRAMDATENSICHERUNG

METHODE DE PROTECTION DE PROGRAMMES

(84) Designated Contracting States:
AT BE CH DE DK ES FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: **22.04.1994 JP 10631694**

(43) Date of publication of application:
10.04.1996 Bulletin 1996/15

(73) Proprietor: **KABUSHIKI KAISYA ADVANCE**
Chuo-ku Tokyo 103 (JP)

(72) Inventors:
- **OTSUKI, Kazunori**
  Yokohama-shi Kanagawa 241 (JP)
- **WATANABE, Shinichirou**
  Kosyatowerkomatsukawa 902
  Tokyo 174 (JP)

(74) Representative: **Thielmann, Andreas**
Cohausz & Florack
Patentanwälte
Postfach 33 02 29
40435 Düsseldorf (DE)

(56) References cited:
JP-A- 1 284 890
JP-A- 3 083 132
JP-A- 4 038 029
JP-A- 63 036 634
JP-A- 63 058 538
JP-A- 63 107 667
US-A- 5 103 476

- **IEICE "1990, IEICE Spring National Convention Lecture Transactions (I)", (1990), p. 282.**

EP 0 706 118 B1

Printed by Jouve, 75001 PARIS (FR)

Description

[0001] The invention relates to a program-data-protecting method according to the methods of claims 1 and 3.

[0002] At present, duplication of data such as application programs, OS software, utility programs, etc. by unauthorized persons is an everyday affair, and no effective countermeasure has yet been devised against unauthorized use of such illegally duplicated programs and software.

[0003] US-A-5,016,276 discloses a common cryptokey generation system and a communication system using common cryptokeys. A center algorithm is applied to identifiers for entities being engaged in communication under a center in order to generate secret algorithms. Further, the secret algorithms are applied to said identifiers for the entities in order to compute common cryptokeys. The generated common cryptokeys are used in a communication system. More specifically in said system the common cryptokeys are used for enciphering plain text before it is transmitted by said communication system to a receiver entity. The receiver entity comprises means for deciphering said received text into plain text again.

[0004] From US-A-5,103,476 a secure system for activating personal computer software at remote locations is known. By providing a registration computer with various information about himself a potential licensee can register to utilize the program. Once the registration process is complete, a tamperproof overlay file is generated at the registration computer and transferred to the personal computer. The tamperproof overlay file includes critical portions of a main program, without which the main program would not operate and also contains licensee identification and license control data. A unique set of encryption and decryption keys is generated and the entire contents of the tamperproof overlay file is encrypted using the encryption key. Based upon the encryption key, a decryption key is provided which is transferred to the computer of the licensee along with said tamperproof overlay file and said main program file now lacking said critical portions.

[0005] It is the object of the invention to provide an improved program-data-protecting method which effectively prevents unauthorized users from using the program data (subsequently mentioned as data).

[0006] That is, the present invention is concerned with a data-protecting system in which a center, for example a vender of data, prepares a special algorithm, i.e., a center algorithm which is secretly held by the center only. The center then prepares a secret algorithm exclusively for the data and for the user by applying the center algorithm to the data, which can include software, for each data to be used by an individual user, and applying the user identifier. The secret algorithm is supplied to the user and to the data or software supplier, and the data or software supplier prepares the shared encryption key used between the data or software supplier and the user relying upon the user identifier and the secret algorithm of the data or software that is supplied, and encrypts part or all of the data or software that is to be supplied directly or indirectly based upon the encryption key, and then supplies it to the user. The user then prepares the shared encryption key used between the data or software supplier and the user, based upon the data or software identifier that is supplied, and his own secret algorithm, and decrypts the encrypted software directly or indirectly.

[0007] Figures 1 to 3 are diagrams illustrating a method of embodying the present invention.

[0008] In the following embodiment, software is referred to as the object to be protected. However, as mentioned above, any data including software can be protected, by the present invention.

[0009] According to the present invention as described above, there is realized a software-protecting system in which a center prepares a special algorithm or a center algorithm which is secretly held by the center only. The center prepares a secret algorithm exclusively used for the software and for the user by applying the center algorithm to the software identifier and the user identifier which are inherent to the software and the user, respectively, and are already known to the public and are used without any substantial change. The secret algorithm is supplied to the user and to the software supplier, and then when it happens that the software supplier must supply software to the user, the software supplier prepares a shared encryption key inherent to both the software that is to be supplied, and the user, by inputting the user identifier into the secret algorithm of the software that is to be supplied, and encrypts part or all of the software that is to be supplied directly or indirectly based on the encryption key and supplies it to the user. The user prepares the shared encryption key between the software that is supplied and the user, by inputting the software identifier that is supplied into his own secret algorithm and decrypts the encrypted software directly or indirectly. Accordingly, the authorized user is allowed to use the software through simple operation but other unauthorized users are not able to use it even though they may be able to copy it.

[0010] That is, according to the present invention, a center (such as, for example, an administration authority) is provided and secretly holds a center algorithm.

[0011] The center prepares a secret algorithm from the center algorithm, user and software identifiers (name, address, administration number, given code, symbols, numerals, etc.), and distributes it to the user and to the software. Note that the identifiers may be one already well-known or not well known to the public or a one inherent to the user or the software which is used without any change, for example. The software to which the center supplies the secret algorithm made for the software is, for example, the software itself, the software supplier, or both.

[0012] Here, the software can be an application program, an OS, a utility program or any other program or data, and the secret algorithm prepared by the center is applied to each one of the software that are to be supplied to the user irrespective of the contents of the software.

[0013] The software supplier can be a supplier that supplies software to the user, such as, a software house, related manufacturer, vendor or software or apparatus for supplying software, or any other entity that supplies software to users requiring a charge or free of charge.

[0014] The software supplier may often be merged into a center, and the center may often be merged with a user. The software supplier could become a user when the supplier takes a position of using the software.

[0015] Here, the user and the software which is designated to be used, will have received a secret algorithm and identifier from the center in advance or just before the operations will be carried out.

[0016] A user means a person who uses the program as well as an apparatus which is directly or indirectly possessed by the user and executes the software, a device associated with the apparatus, the software itself, etc.

[0017] Figure 1 schematically illustrates the operation of the present invention.

[0018] At least a part of a program (P) distributed from the software supplier to the user is encrypted (P') in advance by a separate encryption key, i.e., a second encryption key (K) inherent to the program and a secret algorithm. At the time of installing the program, a user requests the software supplier to supply his identifier (IDu).

[0019] The software supplier prepares a first encryption key by using the identifier (IDu) that is applied and the secret algorithm inherent to the program, encrypts (K') the above-mentioned second encryption key K by using the first encryption key and the encrypted algorithm, and distributes the encrypted second encryption key (K') to the user.

[0020] The user installs the encrypted program (P') by using the encrypted second encryption key (K') that is distributed and installation software that is directly or indirectly attached to the encrypted program (P').

[0021] The installation software prepares a loader which includes the encrypted second encryption key (K') and is linked to the encrypted program (P'). When the loader is executed, the loader always prepares a shared key (first encryption key) by using the user's secret algorithm and the program identifier, and decrypts the encrypted second encryption key (K') together with the decryption algorithm thereby to prepare the second encryption key, and then decrypts (P) the encrypted program (P') by using the second encryption key and the decrypted algorithm.

[0022] In the foregoing was described an indirect method of encrypting or decrypting the program by using two encryption keys. The invention, however, is not limited to the above-mentioned indirect method using a plurality of encryption keys only but can also be applied to a direct method which encrypts or decrypts the program by using a single encryption key (shared key obtained from its own secret algorithm and the user identifier or the program identifier).

[0023] The methods and contents related to steps for preparing shared keys, such as the method of preparing a center algorithm, the method of preparing a secret algorithm, the method of preparing a shared encryption key, entity, definition of identifiers, etc., have been disclosed in Japanese Unexamined Patent Publications (Kokai) Nos. 36634/1988 and 107667/1988, (US-A-5016276).

[0024] The identifiers can be applied to the secret algorithm not only by the systems disclosed in the above-mentioned publications but also by a system disclosed in literature (Matsumoto, Takashima, Imai: "Constitution of Simple One-way Algorithm", Shingakugiho Co., IT89-23, July, 1989).

[0025] The two or more encrypted or decrypted algorithms may be the same ones as represented by, for example, a DES (Data Encryption Standard) system, FEAL (Fast Data Encipherment Algorithm) system, etc. However, any other system may be employed depending upon the speed and the degree of encryption.

Embodiment 1

[0026] Fig. 2 is a diagram for explaining a first embodiment of the present invention. Here, the center portion is the same as the one mentioned above and is not described again.

(1) The user possesses a carrier (e.g., IC card, diskette, or any other storage medium) storing a secret algorithm and a personal authentication algorithm obtained from the center, a carrier execution unit which works in cooperation with the carrier, and an identifier. Similarly, the software supplier possesses a carrier storing an algorithm therein and a carrier execution unit. The software supplier need not possess the algorithm in the constitution of the carrier and the carrier execution unit.

(2) A back-up can be freely executed.

(3) Applicable to all software houses (software suppliers) and to all programs.

Environments and Definitions

[0027] Software house (software supplier): Administers a secret algorithm (program identifier is denoted as IDp) inherent to a program (P) that is to be sold.

[0028] When a program is sold, an enciphered program (P') is sold, which is obtained by enciphering at least a part of the program (P) by utilizing a given random number (K) (second encryption key) (which is inherent to P) and an encryption algorithm. The program

(P') is a file that cannot be executed.

[0029] The user who has purchased the encrypted program (P') applies his own identifier (IDu). Upon receiving an application from the authorized user, therefore, the first encryption key is produced by using the identifier (IDu) and the secret algorithm, and then a random number (K) which is the second encryption key is encrypted by using the first encryption key and the encrypted algorithm to produce an encrypted random number (K') and thereafter the encrypted random number (K') (K' includes data accompanying the first encryption key preparation system) is distributed to the user.

[0030] User: Request the software house to supply his identifier (IDu), at the time to install the purchased program. Sometimes it is not necessary to request to supply. The encrypted random number (K') sent from the software house is input to the installer software. The program is used by using a loader prepared by using the installer software.

[0031] Installation software: Prepares a loader by using an identifier (IDp) input by the user and an encrypted random number (K') and links it to the encrypted program (P'). The installer software is attached to the encrypted program (P') or is separately obtained (distributed free of charge), and is used in common for all programs.

[0032] Loader: Obtains the program (P) by decrypting the encrypted program (P') by using the carrier possessed by the user and the carrier execution unit and by giving, as parameters, an identifier (IDp) of the program possessed in the file and the encrypted random number (K'). The program (P), however, exists in the memory only but does not assume the form of a file. The encrypted program P' is encrypted for the required portion only, and the program (P) does not exist in a complete form. No decryption routine exists in the loader.

[0033] Carrier execution unit: Is a unit which is formed integrally with, separately from, or incorporated in, the target program execution unit (e.g., personal computer, office computer, WS, or any other execution unit) and being connected thereto (using infrared ray, electricity, light, ultrasonic waves, electromagnetic waves, etc.), and is equipped with a mechanism for reading and writing the carrier (e.g., IC card, diskette, or any other recording medium), contains a decryption program (decryption algorithm) (adapter cipher engine: ACE), and decrypts the encrypted program (P') based on the random numbers (K) output by the carrier. The random numbers (K) exist only in the carrier execution unit but are not output to the external unit.

[0034] By taking the future feasibility of this system into consideration, furthermore, it is desired that the ACE is designed to be capable of being version-upgraded or to be capable of being modified (DES → FEAL, etc.). The carrier and the carrier execution unit are only a few examples, and they may further be merged and incorporated into the target program execution unit, or may be formed as an integrated structure, or further may be formed separately from each other, or may be connected additionally or intermediately to an interface connected to a printer or a connecting portion of RS232C or connected to each other, or may be so programmed as to operate in the target program execution unit.

[0035] Further, the carrier execution unit may be an apparatus including a function of the carrier therein, without using a carrier, separately formed from the unit, such as an IC card.

Procedure of Processing

[0036]

(1) Processing on the Software House Side —before the distribution of the program—

- The software house divides the target program (P) into a plurality of loadable modules and, further, so designs the program that the modules are not all loaded at once into the memory.
- The software house encrypts a given part of each of the modules that are divided. Address data of the encrypted part exists in the encrypted program (P'). The address data itself may be encrypted.
- A random number (second encryption key) (K) used for the encryption is unique for each of the programs. It may further be made unique for each of the modules.
- Any encryption means may be used, provided it can be operated by the decryption program (decryption algorithm) ACE incorporated in the carrier execution unit. When the software supplier has its own ACE and distributes it to the users, the encryption means is not necessarily common to all software suppliers.

(2) Processing on the User Side —when the program is purchased— (carrier, carrier execution unit and installer software are assumed to have already been provided)

- The user is registered by the software supplier, and the person identifier is applied.

(3) Processing on the Software House Side —when the user is registered—

- The random number (K) is encrypted (K') by using the identifier (IDu) applied by the user and the secret algorithm (Xp) specific to the program that is distributed.

In this regard, when the secret algorithm (Xp) is used, as shown in Fig. 2, a password code (PIN-P) is input, and a determination of whether or not a

person having the password is the actual registered person, is judged relying upon the personal authentication algorithm (CHA-P). The personal authentication algorithm (CHA-P) and the password code (PIN-P) are provided together with the secret algorithm (Xp) from the center, and may be arbitrarily used and may, further, be arbitrarily provided from the center. The same also holds for the personal authentication algorithm (CHA-U) and the password code (PIN-U) on the user side.

The software supplier sends the encrypted random number (K') to the user. The encrypted random number (K') may be sent by any method such as telephone, facsimile, personal computer communication or floppy disk (when DES is used for encrypting the program P, the amount of data to be sent to the user is, for example, 16 bytes (which corresponds to 32 characters when converted into a character sequence)). The program identifier (IDp) may be notified to the user together with the encrypted random number (K') or may otherwise be printed on the package at the time of distributing the encrypted program (P').

(4) Processing on the User Side —when the program is installed—

- The user starts the installer software and inputs the encrypted random number (K') that is sent and the program identifier (IDp).
- The install software prepares the loader using the encrypted random number (K') that input and the program identifier (IDp), and is linked to the encrypted program (P') (P' with loader). The loader is a utility which can be processed by an OS (MS-DOS), and works as a mediator between the OS and the encrypted program (P'). At this moment, the encrypted program (P') still remains encrypted.

(5) Processing on the User Side —when the program is executed—

- The encrypted program P' with loader is started to authenticate the person who has the carrier.
- The loader prepares the first encryption key (Kup) from the program identifier (IDp) and the secret algorithm (Xu), gives the encrypted random number (K') to the carrier execution unit, and decrypts the encrypted random number (K') based on the first encryption key (Kup) and the decryption program (D). The decrypted random number (K), however, stays in the carrier execution unit and is not output to the external unit.
- The loader gives to the carrier execution unit an encryption part of the encrypted program (P') and the unit decrypts it using the decryption program (DE) and the random number (K), to thereby obtain the program P and thus the program P is executed.
- The loader monitors the execution condition of the program (P) at all times, and causes the carrier execution unit to decrypt the encrypted program (P') every time the encryption portion of the encrypted program (P') is read out.

[0037] In this regard, the encrypted program (P') by itself cannot be decrypted and is delivered to the authorized users only in a variety of states. This may be, for example, a state in which a plurality of programs (the functions of which, however cannot be executed when the password is not given thereto) which already have or will have a secret algorithm are recorded in a large-capacity recording medium such as CD-ROM, and the user who already has or will have the secret algorithm uses the programs and obtains the password and identifier of a program which he likes by paying a royalty.

[0038] In this regard, there is a convenience, as described below, even for the software supplier.

- The software supplier may only prepare the encrypted program by a copying operation, and thus the encrypted program may be mass-produced.
- The hardware that is required can be used by a plurality of software suppliers.

[0039] A further embodiment is illustrated in Fig. 3, wherein a third encryption key, an encryption algorithm and a decryption algorithm are further added to the embodiment of Fig. 2.

[0040] The first encryption key (Kup) is arithmetically obtained by applying the secret algorithm and the identifier (IDp) of the user or program (regarding the user, a target program is the identifier of the program).

[0041] The second encryption key (r) is a random number and is arbitrarily set. The third encryption key (K2) is arbitrarily set in the same manner as the second encryption key.

[0042] The software supplier converts part or all of the third encryption key (K2) into the encrypted third encryption key (K2') using the second encryption key (r) and the encryption algorithm (E2).

[0043] Moreover, the software supplier converts part or all of the second encryption key (r) into the encrypted second encryption key (E(r)) using the first encryption key (Kpu) and the encryption algorithm (E1).

[0044] The software supplier supplies the encryption program (P'), encrypted second encryption key (E(r)) and the encrypted third encryption key (K2') to the user.

[0045] The user prepares the second encryption key (r) which is decrypted from the encrypted second encryption key (E(r)) using the first encryption key (Kup) and the decryption algorithm (D1), and prepares the third encryption key (K2) by decrypting the encrypted third encryption key (K2') using the second encryption key (r) and the decryption algorithm (D2).

[0046] The program (P) is prepared by decrypting the encrypted program (P') using the third encryption key (K2) and the decryption algorithm (D3).

[0047] In the foregoing, the operation of Fig. 3 was described schematically. Other operations are as described with reference to Fig. 2.

[0048] According to the present invention as described above in detail, the software and the user are given specific secret algorithms and an identifier through an authority which is called a center. The user possesses encrypted software and inputs the identifier of the software into his own secret algorithm only when it is desired to easily decrypt it and use it. The operation is thus simple. In addition, possessing the secret algorithm, the user is allowed to use the software as long as the identifier is available even if the software is changed, and thus a burden on the user will be reduced.

[0049] For the unauthorized users, on the other hand, even though they may get it, it is quite difficult for them to decrypt the encrypted software.

Claims

1. A program-data-protecting method in which a center, a program supplier and a user participate, comprising the steps of:

   the center produces a special algorithm held secretly only by the center; applies the special algorithm to a user identifier and a program (P) dividable into a plurality of loadable modules to produce a secret alogrithm (X_p, X_u) exclusive to the program and the user; and supplies the produced secret algorithm to the user and program supplier;

   the program supplier produces a first encryption key (Kpu) using the user identifier (IDu) and the secret algorithm (X_p), encrypts a second encryption key (K) using the first encryption key (Kpu), encrypts one module of the plurality of loadable modules to form an encrypted program (P'), produces an installation program using a program identifier (IDp) and the encrypted second encryption key (K'), wherein the installation program produces, only when the encrypted module in the encrypted program is executed, a common key using the program identifier (IDu) and the secret user algorithm (X_p) to produce a loader for decrypting the encrypted program (P') and supplies the user with the program identifier (IDp), the encrypted second encryption key (K'), the encrypted program (P'), and the installation program; and

   the user causes the supplied installation program to be operated by its own program execution means to cause the installation program to install the encrypted program (P') and produce the loader; operates the loader when the encrypted module of the encrypted program (P') is executed by the program execution means to produce a common key by using the program identifier (IDp) and the secret user algorithm (Xu) to decrypt the encrypted second encryption key (K'); and decrypts the program (P') by using the decrypted second encryption key (K).

2. The program-data-protecting method according to claim 1, wherein said second encryption key (K) is encrypted using a random number.

3. A program data-protecting method in which a center, a program supplier and a user participate, comprising the steps of:

   the center produces a special algorithm held secretly only by the center; applies the special algorithm to a user identifier (IDu) and a program (P) dividable into a plurality of loadable modules to produce a secret algorithm (X_p, X_u) exclusive to the program and the user; and supplies the produced secret algorithm (X_p, X_u) to the user and program supplier;

   the program supplier produces a first encryption key (Kup) using the user identifier (IDu) and the secret algorithm (Xp), encrypts a second encryption key (r) using the first encryption key (Kup); encrypts a third encryption key (K2) using the encrypted second encryption key (r), encrypts one module of the plurality of loadable modules to form an encrypted program (P') using the third encryption key (K2), produces an installation program using the program identifier (IDp) and the encrypted third encryption key (E(r)), wherein the installation program produces, only when the encrypted module in the program is executed, a common key using the program identifier (IDp) and the secret algorithm (Xp) to produce a loader for decrypting the encrypted program (P'); and supplies the user with the program identifier (IDp), the encrypted third encryption key (K2'), the encrypted program (P') and the installation program; and

   the user causes the installation program to be operated by its own program execution means to cause the installation program to install the encrypted program (P') and produce the loader, and to operate the loader when the encrypted module of the encrypted program (P') is executed by the program execution means to produce a common key by using the program iden-

tifier (IDp) and the secret user algorithm (Xu) to decrypt the encrypted second encryption key (E(r)) and decrypt the encrypted third encryption key (K2') and decrypt the encrypted program (P') using the decrypted third encryption key (K2).

Patentansprüche

1. Programmdatenschutzverfahren, an dem eine Zentrale, ein Programmlieferant und ein Benutzer beteiligt sind, umfassend die folgenden Schritte:

   Die Zentrale erzeugt einen speziellen Algorithmus, der geheim, nur von der Zentrale verwahrt wird; wendet den speziellen Algorithmus auf eine Benutzerkennung und ein Programm (P), das in eine Mehrzahl ladbarer Module geteilt werden kann, an, um einen geheimen Algorithmus (Xp, Xu) zu erzeugen, der dem Programm und dem Benutzer eigen ist; und liefert den erzeugten geheimen Algorithmus an den Benutzer und den Programmlieferanten;

   der Programmlieferant erzeugt einen ersten Verschlüsselungsschlüssel (Kpu) durch Verwenden der Benutzerkennung (IDu) und des geheimen Algorithmus (Xp), verschlüsselt einen zweiten Verschlüsselungsschlüssel (K) durch Verwenden des ersten Verschlüsselungsschlüssels (Kpu), verschlüsselt ein Modul der Mehrzahl ladbarer Module, um ein verschlüsseltes Programm (P') zu bilden, erzeugt ein Installationsprogramm durch Verwenden einer Programmkennung (IDp) und des verschlüsselten zweiten Verschlüsselungsschlüssels (K'), wobei das Installationsprogramm, nur wenn das verschlüsselte Modul im verschlüsselten Programm ausgeführt wird, durch Verwenden der Programmkennung (IDu) und des geheimen Benutzeralgorithmus (Xp) einen gemeinsamen Schlüssel erzeugt, um einen Programmlader zum Entschlüsseln des verschlüsselten Programmes (P') zu erzeugen, und liefert dem Benutzer die Programmkennung (IDp), den verschlüsselten zweiten Verschlüsselungsschlüssel (K'), das verschlüsselte Programm (P') und das Installationsprogramm; und

   der Benutzer bewirkt, daß das gelieferte Installationsprogramm von seinem eigenen Programmausführungsmittel betrieben wird, um zu veranlassen, daß das Installationsprogramm das verschlüsselte Programm (P') installiert und den Programmlader erzeugt; den Programmlader betreibt, wenn das verschlüsselte Modul des verschlüsselten Programmes (P') durch das Programmausführungsmittel ausgeführt wird, um einen gemeinsamen Schlüssel durch Verwenden der Programmkennung (IDp) und des geheimen Benutzeralgorithmus (Xu) zu erzeugen, um den verschlüsselten zweiten Verschlüsselungsschlüssel (K') zu entschlüsseln; und entschlüsselt das Programm (P') durch Verwenden des entschlüsselten zweiten Verschlüsselungsschlüssels (K).

2. Programmdatenschutzverfahren nach Anspruch 1, wobei der zweite Verschlüsselungsschlüssel (K) durch Verwenden einer Zufallszahl verschlüsselt wird.

3. Programmdatenschutzverfahren, an dem eine Zentrale, ein Programmlieferant und ein Benutzer beteiligt sind, umfassend die folgenden Schritte:

   Die Zentrale erzeugt einen speziellen Algorithmus, der geheim, nur von der Zentrale verwahrt wird; wendet den speziellen Algorithmus auf eine Benutzerkennung (IDu) und ein Programm (P), das in eine Mehrzahl ladbarer Module geteilt werden kann, an, um einen geheimen Algorithmus (Xp, Xu) zu erzeugen, der dem Programm und dem Benutzer eigen ist; und liefert den erzeugten geheimen Algorithmus (Xp, Xu) an den Benutzer und den Programmlieferanten;

   der Programmlieferant erzeugt einen ersten Verschlüsselungsschlüssel (Kpu) durch Verwenden der Benutzerkennung (IDu) und des geheimen Algorithmus (Xp), verschlüsselt einen zweiten Verschlüsselungsschlüssel (r) durch Verwenden des ersten Verschlüsselungsschlüssels (Kpu); verschlüsselt einen dritten Verschlüsselungsschlüssel (K2) durch Verwenden des verschlüsselten zweiten Verschlüsselungsschlüssels (r), verschlüsselt durch Verwenden des dritten Verschlüsselungsschlüssels (K2) ein Modul der Mehrzahl ladbarer Module, um ein verschlüsseltes Programm (P') zu bilden, erzeugt ein Installationsprogramm durch Verwenden der Programmkennung (IDp) und des verschlüsselten dritten Verschlüsselungsschlüssels (E(r)), wobei das Installationsprogramm, nur wenn das verschlüsselte Modul im Programm ausgeführt wird, durch Verwenden der Programmkennung (IDu) und des geheimen Algorithmus (Xp) einen gemeinsamen Schlüssel erzeugt, um einen Programmlader zum Entschlüsseln des verschlüsselten Programmes (P') zu erzeugen; und liefert dem Benutzer die Programmkennung (IDp), den verschlüsselten dritten Ver-

schlüsselungsschlüssel (K2'), das verschlüsselte Programm (P') und das Installationsprogramm; und

der Benutzer bewirkt, daß das gelieferte Installationsprogramm von seinem eigenen Programmausführungsmittel betrieben wird, um zu veranlassen, daß das Installationsprogramm das verschlüsselte Programm (P') installiert und den Programmlader erzeugt, und den Programmlader betreibt, wenn das verschlüsselte Modul des verschlüsselten Programmes (P') durch das Programmausführungsmittel ausgeführt wird, um einen gemeinsamen Schlüssel durch Verwenden der Programmkennung (IDp) und des geheimen Benutzeralgorithmus (Xu) zu erzeugen, um den verschlüsselten zweiten Verschlüsselungsschlüssel (E(r)) zu entschlüsseln und den verschlüsselten dritten Verschlüsselungsschlüssel (K2') zu entschlüsseln und das verschlüsselte Programm (P') durch Verwenden des entschlüsselten dritten Verschlüsselungsschlüssels (K2) zu entschlüsseln.

Revendications

1. Méthode de protection de programmes-données à laquelle participent un centre, un fournisseur de programme et un utilisateur, comprenant les étapes suivantes :

   le centre produit un algorithme spécial maintenu secret par le centre uniquement ; applique l'algorithme spécial à un identificateur d'utilisateur et un programme (P) divisible en une pluralité de modules chargeables afin de produire un algorithme secret (X_p, X_u) exclusif au programme et à l'utilisateur ; et fournit l'algorithme secret produit à l'utilisateur et au fournisseur de programme ;
   le fournisseur de programme produit une première clé de chiffrement (Kpu) utilisant l'identificateur d'utilisateur (IDu) et l'algorithme secret (X_p), crypte une seconde clé de chiffrement (K) en utilisant la première clé de chiffrement (Kpu), crypte un module de la pluralité de modules chargeables pour former un programme chiffré (P'), produit un programme d'installation utilisant un identificateur de programme (IDp) et la seconde clé de chiffrement cryptée (K'), le programme d'installation produisant, uniquement lorsque le module crypté dans le programme crypté est exécuté, une clé commune utilisant l'identificateur de programme (IDu) et l'algorithme d'utilisateur secret (X_p) pour produire un chargeur pour décrypter le programme crypté (P') et fournit à l'utilisateur l'identificateur de programme (IDp), la seconde clé de chiffrement cryptée (K'), le programme crypté (P') et le programme d'installation ; et
   l'utilisateur commande la mise en oeuvre du programme d'installation fourni par le moyen d'exécution de programme propre à ce dernier pour faire en sorte que le programme d'installation installe le programme crypté (P') et produise le chargeur ; l'utilisateur met en oeuvre le chargeur lorsque le module crypté du programme crypté (P') est exécuté par le moyen d'exécution de programme afin de produire une clé commune en utilisant l'identificateur de programme (IDp) et l'algorithme d'utilisateur secret (Xu) pour décrypter la seconde clé de chiffrement cryptée (K') ; et décrypte le programme (P') en utilisant la seconde clé de chiffrement décryptée (K).

2. Méthode de protection de programmes-données selon la revendication 1, dans laquelle ladite seconde clé de chiffrement (K) est cryptée en utilisant un nombre aléatoire.

3. Méthode de protection de programmes-données à laquelle participent un centre, un fournisseur de programme et un utilisateur, comprenant les étapes suivantes :

   le centre produit un algorithme spécial maintenu secret par le centre uniquement ; applique l'algorithme spécial à un identificateur d'utilisateur (IDu) et un programme (P) divisible en une pluralité de modules chargeables afin de produire un algorithme secret (X_p, X_u) exclusif au programme et à l'utilisateur ; et fournit l'algorithme secret produit (X_p, X_u) à l'utilisateur et au fournisseur de programme ;
   le fournisseur de programme produit une première clé de chiffrement (Kup) utilisant l'identificateur d'utilisateur (IDu) et l'algorithme secret (Xp), crypte une seconde clé de chiffrement (r) utilisant la première clé de chiffrement (Kup) ; crypte une troisième clé de chiffrement (K2) utilisant la seconde clé de chiffrement cryptée (r), crypte un module de la pluralité de modules chargeables pour former un programme crypté (P') utilisant la troisième clé de chiffrement (K2), produit un programme d'installation utilisant l'identificateur de programme (IDp) et la troisième clé de chiffrement cryptée (E(r)), le programme d'installation produisant, uniquement lorsque le module crypté dans le programme est exécuté, une clé commune utilisant l'identificateur de programme (IDp) et l'algorithme secret (Xp) pour produire un chargeur afin de décrypter le programme crypté (P') ; et fournit à l'utilisateur l'identificateur de program-

me (IDp), la troisième clé de chiffrement cryptée (K2'), le programme crypté (P') et le programme d'installation ; et

l'utilisateur commande la mise en oeuvre du programme d'installation par le moyen d'exécution de programme propre à ce dernier afin de faire en sorte que le programme d'installation installe le programme crypté (P') et produise le chargeur, et de mettre en oeuvre le chargeur lorsque le module crypté du programme crypté (P') est exécuté par le moyen d'exécution de programme afin de produire une clé commune en utilisant l'identificateur de programme (IDp) et l'algorithme utilisateur secret (Xu) pour décrypter la seconde clé de chiffrement cryptée (E(r)) et décrypter la troisième clé de chiffrement cryptée (K2') et décrypter le programme crypté (P') utilisant la troisième clé de chiffrement décryptée (K2).

Fig.1

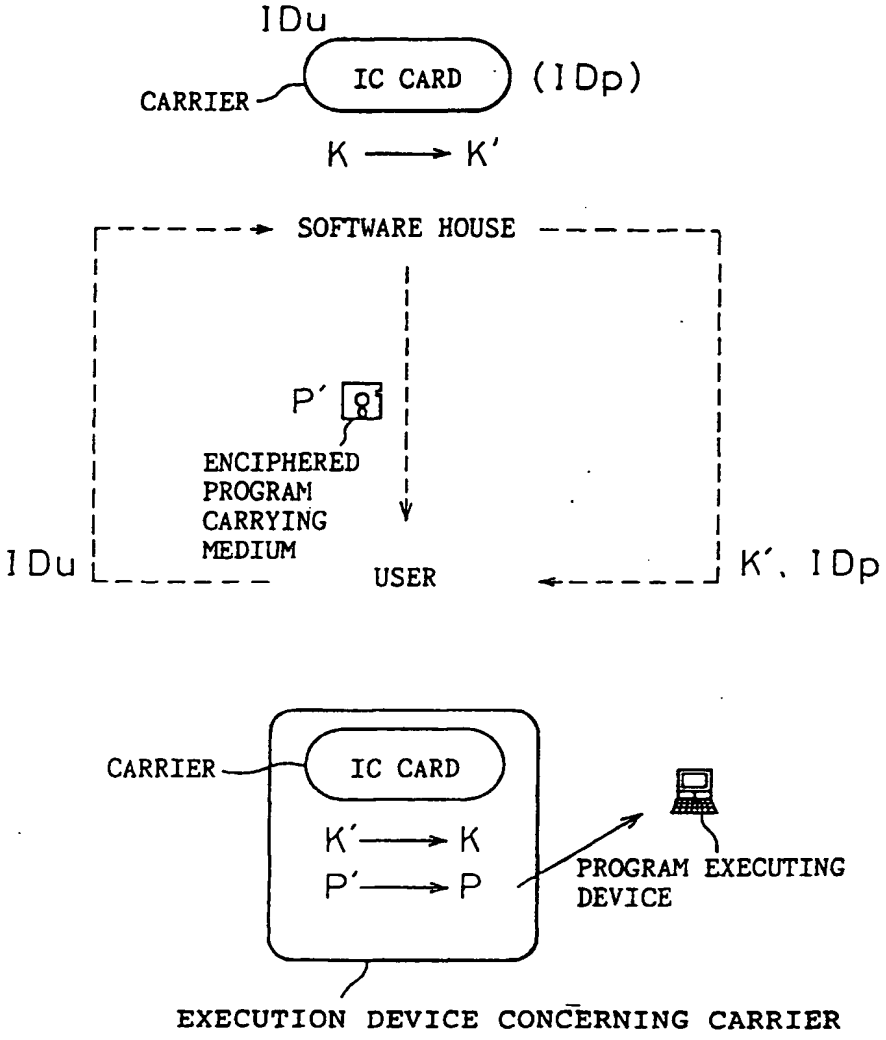

IDu
CARRIER
IC CARD
(IDp)
K ⟶ K'
SOFTWARE HOUSE
P'
ENCIPHERED PROGRAM CARRYING MEDIUM
IDu
USER
K', IDp
CARRIER
IC CARD
K' ⟶ K
P' ⟶ P
PROGRAM EXECUTING DEVICE
EXECUTION DEVICE CONCERNING CARRIER

Fig. 2

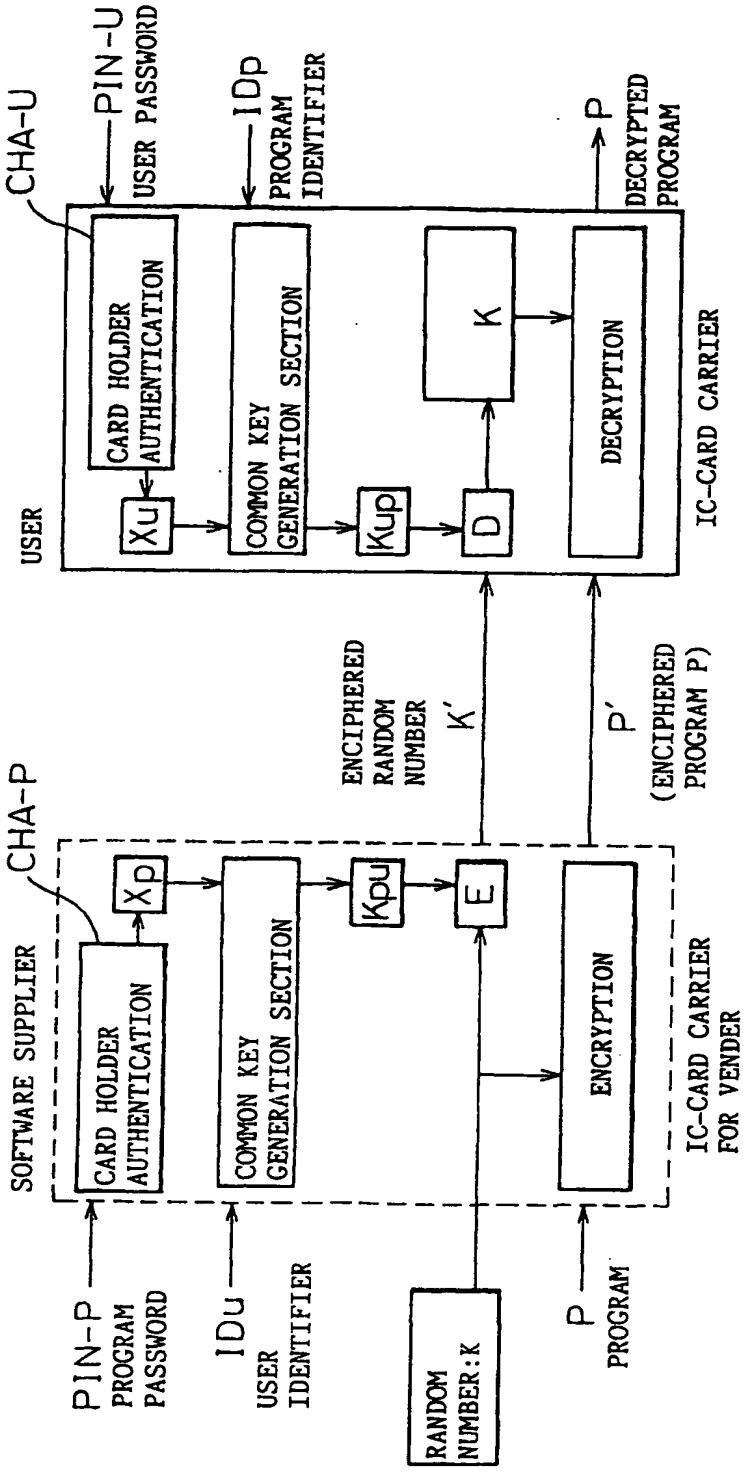
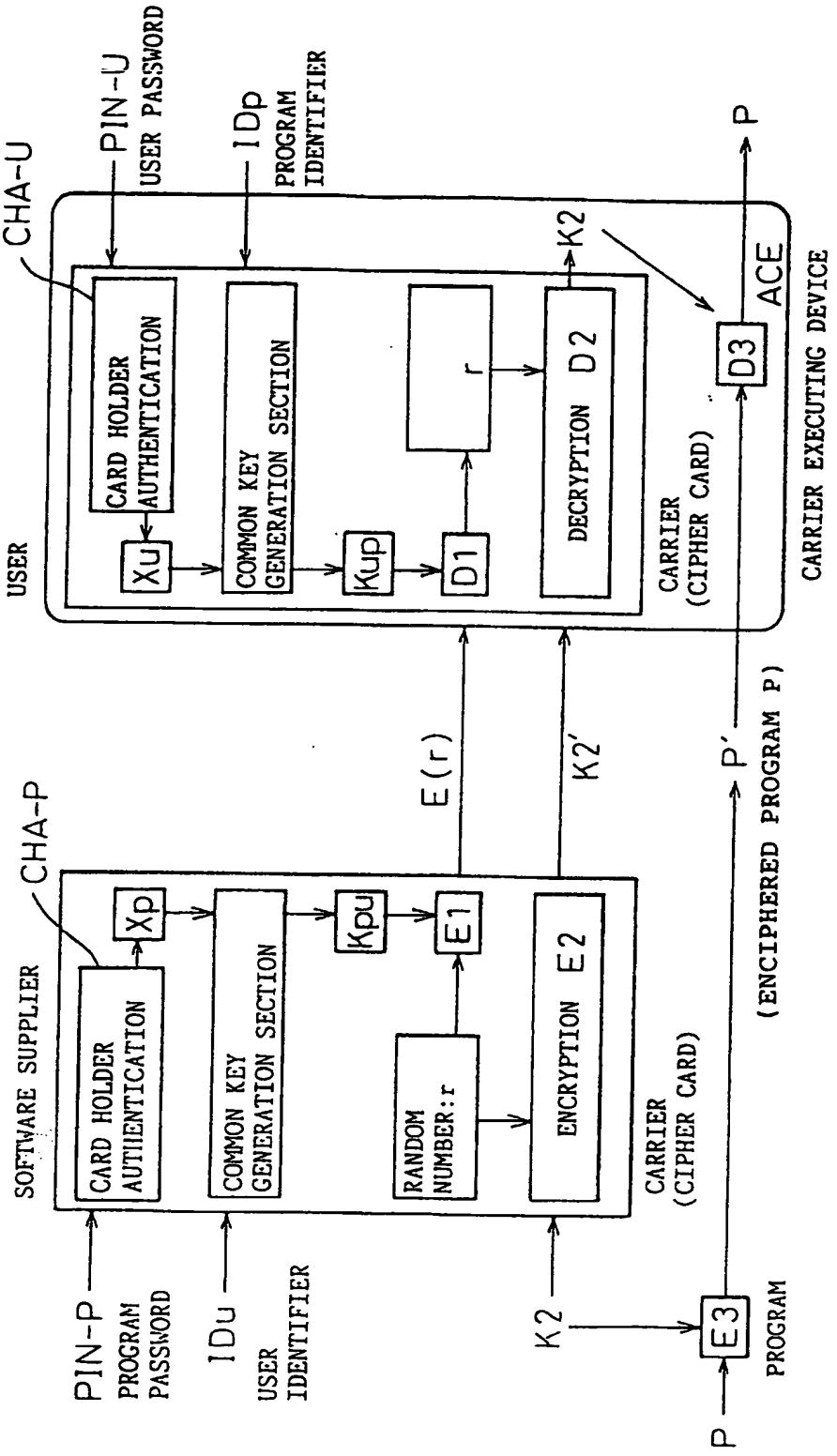

CHA-U
PIN-U
USER PASSWORD
IDp
PROGRAM IDENTIFIER
P
DECRYPTED PROGRAM
USER
CARD HOLDER AUTHENTICATION
Xu
COMMON KEY GENERATION SECTION
Kup
D
K
DECRYPTION
IC-CARD CARRIER
ENCIPHERED RANDOM NUMBER
K'
P'
(ENCIPHERED PROGRAM P)
CHA-P
SOFTWARE SUPPLIER
Xp
CARD HOLDER AUTHENTICATION
COMMON KEY GENERATION SECTION
Kpu
E
ENCRYPTION
IC-CARD CARRIER FOR VENDER
PIN-P
PROGRAM PASSWORD
IDu
USER IDENTIFIER
RANDOM NUMBER:K
P
PROGRAM

Fig.3


CHA-U
PIN-U
USER PASSWORD
IDp
PROGRAM IDENTIFIER
USER
CARD HOLDER AUTHENTICATION
Xu
COMMON KEY GENERATION SECTION
Kup
D1
r
DECRYPTION D2
K2
CARRIER (CIPHER CARD)
D3
ACE
CARRIER EXECUTING DEVICE
P
E(r)
K2'
P'
(ENCIPHERED PROGRAM P)
SOFTWARE SUPPLIER
CHA-P
Xp
CARD HOLDER AUTHENTICATION
COMMON KEY GENERATION SECTION
Kpu
E1
RANDOM NUMBER:r
ENCRYPTION E2
CARRIER (CIPHER CARD)
PIN-P
PROGRAM PASSWORD
IDu
USER IDENTIFIER
K2
E3
P
PROGRAM